REMARKS

This is in response to the Office Action dated July 27, 2007 in which claims 1-16 and 19-24 were rejected. Applicant respectfully requests reconsideration and allowance of all pending claims in view of the above-amendments and the following remarks.

I. CLAIM AMENDMENTS

Applicant requests that the above-claim amendments be entered for the following reasons:

A. **Newly Asserted §101 Rejections**

Claims 1 and 24 are amended in response to a §101 rejection that was not asserted with the first Office Action, but newly asserted in the final Office Action. Since Applicant must be given an opportunity to respond to the rejection, Applicants request that the proposed amendments be entered.

Further, the proposed amendments simply add that the claimed subject matter is "stored on a computer readable memory", which does not raise new issues that would require further search an/or consideration.

B. **Amendments to Claims 1 and 24 – Previously Examined Limitations**

Claim 1 is amended to include elements from dependent claim 2 ("at run time") and from existing independent claim 8 ("building", "assigning" and "displaying").

Claim 24 is amended to include elements from existing independent claim 8 ("build" and "assign").

Since these newly added elements are already present in existing independent claim 8 and have already been examined by the Examiner, Applicant respectfully requests that the proposed amendments be entered.

If the Examiner finds that the proposed amendments do not place the claims in condition for allowance, Applicant requests that the amendments be entered to place the claims in better condition for appeal.

C. **Remaining Amendments – Formality Objections**

The remaining amendments are in response to minor formality objections regarding simple typographical errors. These amendments are also believed not to raise new issues.

II. CLAIM OBJECTIONS

Claims 7, 8 and 23 were objected to for minor typographical errors. These claims are amended as suggested in the Office Action.

Accordingly, Applicant requests that these objections be withdrawn.

III. CLAIM REJECTIONS UNDER §101

Claims 1-7 and 24 were rejected under §101 because the claimed invention is directed to allegedly non-statutory subject matter.

While Applicant respectfully disagrees, claims 1 and 24 are amended to recite that the command processor (in claim 1) and the software design suite (in claim 24) are "stored on a computer readable memory".

According to the Interim Guidelines for Examination of Patent Applications for Patent Subject Matter Eligibility (Official Gazette, November 22, 2005), when functional descriptive material is recorded on some computer-readable medium it becomes structurally and functionally interrelated to the medium and will be statutory in most cases since use of technology permits the function of the descriptive material to be realized.

Claims 1-7 and 24 are therefore statutory.

IV. CLAIM REJECTIONS UNDER §102(b)

Claims 1-4, 7-13 and 19-20 were rejected under §102(b) as being anticipated by Nahaboo et al., U.S. Patent No. 5,974,253.

The Examiner misinterprets the Nahaboo et al. reference as applied to the elements of Applicant's claims.

A. **Nahaboo et al.**

Nahaboo et al. describe a system having an edit mode and a separate execution mode. (Column 6, lines 60-64).

Nahaboo et al. contains an editor (described in Column 2, lines 43-56) that allows the user to create, arrange or modify a preexisting, hard-coded set of graphical objects, called "widgets", that are selected from a "widget container" using a menu. Using this technique, the time to change the user interface configuration may range form a few minutes to a few hours.

The graphical objects cannot be modified during the execution mode.

Nahaboo et al. describe in column 3, lines 53-56,

> "In order to function, this interactive description interface (10) must be associated with a library of interactive command objects (called command object toolbox "X/MOTIF" (20) in the UNIX environment) and a graphical object toolbox that includes a library of graphical objects "GO" (21)." (Emphasis added).

In other words, the commands and objects are pre-existing and hard-coded. Column 9, lines 36-42 state,

> "Selecting the 'geometry' menu, using button 3111, will activate an editing function that allows the user to display selection and modification attributes of a widget's geometry. The "behavior" menu, activated using button 3112, allows the user to activate an editing function that will display the attribute that defines a widget's behavior and what its behavior will be when using the attributes selected." (Emphasis added).

This manual, on-screen selection and modification process can therefore be very time-consuming and not easily changed from one session to the next or between users. A new user to the same session would therefore need to repeat each button selection and editing operation described above for each widget, etc. the is desired to be changed.

Nahaboo et al. describe that, "We have used the word "script" to designate the code that describes the response to the set of callbacks". (Col. 11, lines 55-56) These callbacks define the responses to actions such as selection of a pushbutton. They do not correspond to configuration commands that modify the graphical user interface at run time of the graphical user interface. Rather the function defined by the callback procedure can be edited only in the edit mode (Col. 9, line 47 to col. 10, line 2). Once the edit mode has terminated, the function cannot be edited at run time during the execution mode.

Further, Nahaboo et al. have a table (described in column 4, line 63 and in Fig 1B) that stores the association between a graphical object a selector and a call back in the interpreter. These are fixed and predetermined when the tool is invoked during execution mode.

During execution mode of the tool, selecting an object will activate the stored function This function determines the action performed when the user selects an object, which can be changed by

the user only using the editor (in edit mode), and not during execution mode.

In fact, Nahaboo et al. state that a backup of the interface can be made using the "Save" command from the "file" menu. "Afterwards, the saved file can be loaded using the "Load" command." (Column 13, lines 32-35).

The saved backup merely saves an interface that has already been configured (i.e., a modified matrix array) (Col. 1, lines 56-60 and col. 4, line 67 to col. 5, line 4). When reloading the saved interface with the "Load" command, the saved interface does not contain "configuration commands" that are loaded into a command interpreter and modify a graphical user interface by building objects or assigning functionality to the built objects. Rather, the configuration is already fixed and predefined by the saved file. The configuration cannot be modified at run time during execution of the tool.

B. **Independent Claim 1**

Claim 1 requires "a command interpreter, which loads one or more configuration commands into the command processor." The command processor interprets the configuration commands and modifies the graphical user interface at run time of the graphical user interface according to the interpreted configuration commands.

Nahaboo et al. have two mode, an edit mode and a separate execution mode. Nahaboo et al. do not enable modifying a graphical user interface at run time of that interface.

Further, Nahaboo et al. do not disclose performing such modifications through configuration commands that are interpreted by a command processor. Rather, Nahaboo et al. use a menu and selection method to select from pre-defined objects and functions.

Nahaboo et al. do not disclose loading "configuration commands into the command processor from at least one of a user specified command configuration script comprising the one or more configuration commands," as recited in claim 1. Again, modifications are performed through menu selection. The "script" mentioned by Nahaboo et al. define the responses to actions such as selection of a pushbutton. They do not correspond to configuration commands that modify the graphical user interface at run time of the interface.

Similarly, Nahaboo et al. do not disclose loading "configuration commands into the

command processor . . . from a command line . . . entered by the user," as recited in claim 1, wherein those configuration commands modify the graphical user interface at run time of the interface.

Further, Nahaboo et al. do not disclose such configuration commands (loaded at run time) that modify the interface by:

> "building graphical objects according to the interpreted configuration commands;
> assigning functionality to the built graphical objects according to the interpreted configuration commands; and
> displaying a user-interactive window containing the graphical objects according to the interpreted configuration commands."

Rather, the widgets of Nahaboo et al. are edited by a separate editor (not during execution mode). Once configured the interface can be loaded and executed. But at that point, the objects have already been configured previously by the editor and are not further modified during execution.

The "Save" and "Load" commands referred to by Nahaboo et al. in column 13, lines 32-35 do not meet the limitations of claim 1. The saved backup merely saves an interface that has already been configured (i.e., a modified matrix array) (Col. 1, lines 56-60 and col. 4, line 67 to col. 5, line 4). When reloading the saved interface with the "Load" command, the saved interface does not contain "configuration commands" that are loaded into a command interpreter and modify a graphical user interface by "building" objects or "assigning functionality to the built objects". Rather, the configuration is already fixed and predefined by the saved file. The configuration cannot be modified at run time during execution of the tool.

For at least the above reasons, the Nahaboo et al. patent does not anticipate the elements of claim 1.

Applicants respectfully request that the rejection of claim 1 and its dependent claim 7 under §102(b) be withdrawn.

B. **Independent Claim 8**

For similar reasons, Nahaboo et al. also do not anticipate the elements of independent claim

8.

In addition, claim 8 recites,

“”upon execution of a command processor, loading a top level Tool Command Language (TCL) command into a namespace;

loading one or more TCL commands ...

building graphical objects according to the TCL configuration commands;

assigning functionality to the built graphical objects according to the TCL configuration commands; and

displaying a user-interactive window containing the graphical objects according to the TCL configuration commands.”

As described above, Nahaboo et al. do not allow for objects to be built and functionality to be assigned in the manner recited in claim 8.

Rather, the objects are selected from a pre-defined menu by a separate editor, not through configuration commands loaded through a command processor upon execution of the command processor. If the particular interface is saved as a backup and then loaded, the loaded file does not include a script containing configuration commands that are loaded into the command processor and serve to build graphical objects or assign functionality according to such commands.

In contrast, when reloaded, the interface contains pre-defined objects and functionality. Nahaboo et al. also do not provide for configuration commands to be loaded by the user through a command line during execution.

The Nahaboo et al. patent does not anticipate the elements of claim 8, and Applicant respectfully requests that the rejection of claim 8 and its dependent claims 9-13 under §102(b) be withdrawn.

C. **Independent Claim 19**

Independent claim 19 includes similar limitations as claims 1 and 8 that are not anticipated by Nahaboo et al.

In addition, claim 19 recites,

“assembling a graphical user interface having no hard coded objects based on the

interpreted configuration commands from the user;

wherein all objects within the graphical user interface are user defined through the one or more configuration commands." (Emphasis added).

The Office Action cites column 6, lines 53-59 of Nahaboo et al. as somehow supporting "assembling a graphical user interface having no hard coded objects based on the interpreted configuration commands from the user, as recited in claim 19.

Nahaboo et al. expressly state,

> "In order to function, this interactive description interface (10) must be associated with a library of interactive command objects . . . and a graphical object toolbox that includes a library of graphical objects . . .." (Emphasis added).

The objects are clearly pre-existing and hard-coded. The user must select an object from a "widget container" and then alter its characteristics through a manual menu selection process.

The Office Action further states, that, "the fact that the user can enter the 'editing' mode without entering the 'execution' mode means that said user can define all objects of the graphical user interface before execution, thus defining all objects of a graphical user interface having no hard coded objects."

The Examiner seems to ignore the context of claim 19 in which the objects are defined through configuration commands that are loaded into a command interpreter through a script of such commands or a command line. All objects are defined through this process, and the interface has no hard-coded objects.

If fact, the Examiner's statement emphasizes that all objects are defined before execution. These objects are not defined during execution of a command processor through configuration commands loaded into the processor through a script or a command line. Besides, the objects are hard-coded, contrary to claim 19.

The Nahaboo et al. patent does not anticipate the elements of claim 19, and Applicant respectfully requests that the rejection of claim 19 and its dependent claim 20 under §102(b) be withdrawn.

V. CLAIM REJECTIONS UNDER §103(a)

Claims 5-6, 14-16 and 21-24 were rejected under §103(a) as being unpatentable over Nahaboo et al. and Dangelo et al., U.S. Patent No. 5,493,508.

A. **Dependent Claims 5-6, 14-16 and 21-22**

Dangelo was cited merely for disclosing a suite of integrated circuit design tools.

Neither Nahaboo et al. nor Dangelo disclose the elements discussed above with respect to independent claims 1, 8 and 19. Thus, even if Nahaboo et al. were modified as suggested in the Office Action, the resulting modification would still lack these elements.

B. **Independent Claim 24**

Claim 24 states that the IC design tools operate under control of the command processor and within the graphical user interface.

The graphical user interface is "specified entirely by a user through one or more configuration commands loaded into the command processor at run time of the command processor."

In Nahaboo et al., the interface is specified entirely before run time of a design tool.

Further, Nahaboo et al. do not disclose configuration commands that, at run time, build graphical objects for the graphical user interface and assign functionality to the built graphical objects. The objects are not built through such commands and are not built at run time as defined in claim 24.

These elements would not be obvious to a person of ordinary skill in the art since Nahaboo et al. disclose such a different method for altering an interface. Nahaboo et al. require selection from predefined objects and functions, such as from a table of functions or attributes using a separate editor during an edit mode.

Since neither Nahaboo et al. nor Dangelo disclose the elements of claim 24, and Applicant respectfully requests that the rejection of claim 24 under §103(a) be withdrawn.

The Director is authorized to charge any fee deficiency required by this paper or credit any overpayment to Deposit Account No. 12-2252.

Respectfully submitted,

WESTMAN, CHAMPLIN & KELLY, P.A.

By: /David D. Brush/

David D. Brush, Reg. No. 34,557
900 Second Avenue South, Suite 1400
Minneapolis, Minnesota 55402-3319
Phone: (612) 334-3222 Fax: (612) 334-3312

DDB:tkj